Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A device comprising:

a status register to store a word including a plurality of sets of arithmetic flags, each set having M bits and associated with one of a plurality of data items of varying field sizes ~~and having M bits~~ and wherein the status register can store sets of varying field size and the word is a single instruction multiple data (SIMD) word;

a combination function module to examine the word stored in the status register to determine a data item field size of each set of arithmetic flags for the word, and based on the determination of the data item field size to logically combine the plurality of arithmetic flags of the sets within the word into a single combined arithmetic flag variable of M bits, and to store the single combined arithmetic flag variable into a second register, wherein, in combining, the combination function module performs an OR operation;

wherein each of the plurality of arithmetic flags ~~represent~~ represents a result status of ~~the plurality of data items after~~ a mathematical operation [[is]] performed by a processor [[on]] to obtain one of the plurality of data items; and

a condition check module to determine the result status of the combined arithmetic flag variable and cause the processor to execute an appropriate operation based on the result status.

Claim 2 (canceled)

Claim 3 (previously presented): The device recited in claim 1, wherein the field size is based on a nibble, byte, half word, or word in length.

Claim 4 (original): The device recited in claim 3, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

Claim 5 (canceled)

Claim 6 (previously presented): The device recited in claim 1, wherein the result status determined by the condition check module further comprises:

any data item has overflowed;

any data item has not overflowed;

any data item is positive or zero;

any data item is negative;

any data item is zero;

any data item is not zero;

any data item has a carry out;

any data item does not have a carry out;

all data items have overflowed;

all data items have not overflowed;

all data items are positive or zero;

all data items are negative;

all data items are zero;

all data items are not zero;

all data items have a carry out; and

all data items do not have a carry out.

Claim 7 (currently amended): A method comprising:

determining a field size of each of a plurality of sets of arithmetic flags stored in a status register of a processor on which to base a combination process, wherein each set of the plurality of arithmetic flags ~~represent~~ represents a result status of a ~~data item of a plurality of data items after a~~ mathematical operation [[is]] performed by the processor [[on]] to obtain the data item, wherein the word is a single instruction multiple data (SIMD) word;

extracting the plurality of sets of arithmetic flags from the status register based on the field size;

logically combining the plurality of sets of arithmetic flags based on a function selected when a combination process is selected, wherein the function comprises an OR operation; and

storing a result of the combining of the plurality of arithmetic flags in a destination register for access by the processor.

Claim 8 (previously presented): The method recited in claim 7, wherein the field size is based on a nibble, byte, half word, or word in length.

Claim 9 (original): The method recited in claim 8, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

Claim 10 (canceled)

Claim 11 (previously presented): The method recited in claim 7, wherein the function may be used to determine the result status of the plurality of data items, said status comprising:

any data item has overflowed;

any data item has not overflowed;

any data item is positive or zero;

any data item is negative;

any data item is zero;

any data item is not zero;

any data item has a carry out;

any data item does not have a carry out;

all data items have overflowed;

all data items have not overflowed;

all data items are positive or zero;

all data items are negative;

all data items are zero;

all data items are not zero;

all data items have a carry out; and

all data items do not have a carry out.

Claim 12 (currently amended): An apparatus comprising a data storage medium for storing instructions, wherein the instructions, when executed by a processor, result in the processor performing a method, the method comprising:

determining a field size of each of a plurality of sets of arithmetic flags stored in a status register of a processor on which to base a combination process, wherein each set of the plurality of arithmetic flags ~~represent~~ represents a result status of a ~~data item of a plurality of data items after a~~ mathematical operation [[is]] performed by the processor [[on]] to obtain the data item;

extracting the plurality of sets of arithmetic flags from the status register based on the field size;

logically combining the plurality of sets of arithmetic flags based on a function selected when a combination process is selected, wherein the function comprises an OR operation; and

storing a result of the combining of the plurality of arithmetic flags in a destination register for access by the processor.

Claim 13 (previously presented): The apparatus recited in claim 12, wherein the field size is based on a nibble, byte, half word, or word in length.

Claim 14 (original): The apparatus recited in claim 13, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

Claim 15 (canceled)

Claim 16 (previously presented): The apparatus recited in claim 12, wherein the function may be used to determine the result status of the plurality of data items, said result status comprising:

any data item has overflowed;
any data item has not overflowed;
any data item is positive or zero;
any data item is negative;
any data item is zero;
any data item is not zero;
any data item has a carry out;
any data item does not have a carry out;
all data items have overflowed;
all data items have not overflowed;
all data items are positive or zero;
all data items are negative;
all data items are zero;
all data items are not zero;
all data items have a carry out; and
all data items do not have a carry out.

Claims 17 – 22 (canceled)

Claim 23 (currently amended): A system comprising:

a processor having a first register to store a word including a plurality of sets of arithmetic flags, each set having M bits and associated with one of a plurality of data items of varying field sizes ~~and having M bits~~, and a second register to store a single combined arithmetic flag variable of M bits;

a combination function module to examine the word to determine a data item field size of each set of arithmetic flags for the word, and based on the determination of the data item field size to logically combine the plurality of arithmetic flags of the sets within the word into the single combined arithmetic flag variable, wherein, in combining, the combination function module performs an OR operation; and wherein each of the plurality of arithmetic flags ~~represent~~

represents a result status of ~~the plurality of data items after~~ a mathematical operation [[is]] performed by the processor [[on]] to obtain one of the plurality of data items; and

the processor including a condition check module coupled to the combination function module, the processor to receive the single combined arithmetic flag variable and to determine the next operation to perform based upon the result status of the single combined arithmetic flag variable.

Claim 24 (previously presented): The system of claim 23, wherein the processor includes at least three stages of pipelining.

Claim 25 (previously presented): The system of claim 24, wherein the at least three stages of pipelining include a fetch stage, a decode stage, and an execute stage.